Wissenschaftliches Arbeiten unter einer grafischen Oberfläche

Manhart, Klaus

Veröffentlichungsversion / Published Version
Zeitschriftenartikel / journal article

Wissenschaftliches Arbeiten unter einer grafischen Oberfläche

von Klaus Manhart [1]

Zusammenfassung

Eine neue Generation von PC-Betriebssystemen mit grafischen, fensterorientierten Oberflächen erlaubt ein komfortables, kreatives und zeitsparendes wissenschaftliches Arbeiten auf billiger Massenhardware. Am Beispiel von Windows werden die wesentlichen Vorzüge und Möglichkeiten grafischer Oberflächen anhand von konkreten Problemstellungen illustriert: Texterfassung, die gleichzeitige Darstellung unterschiedlicher Informationsquellen, der bequeme Datenaustausch mit und ohne OLE-Verfahren und das simultane Ablaufenlassen von Programmen. Wir stützen uns dabei auf Erfahrungen bei der Erstellung einer Dissertation, ein Vergleich oder eine grundlegende Bewertung ist nicht beabsichtigt. Der Artikel soll als Anregung verstanden werden, die Leistungsfähigkeit fensterorientierter Betriebssysteme beim wissenschaftlichen Arbeiten zu nutzen.

Abstract

A new generation of PC operating systems using graphical user interfaces gives a researcher a comfortable, creative and time-saving working environment on cheap mass-produced hardware. MS-Windows embodies the essential advantages and features of a graphical interface: high-end word-processing, the presentation of different information at the same time, comfortable data-exchange and the simultaneous execution of programs. In doing this, we examine how MS-Windows was used in writing my dissertation. There is no comparison or evaluation. This article should be an incentive to use the power of graphical user interfaces in academic work.

1 *Klaus Manhart* ist wissenschaftlicher Mitarbeiter am Institut für Soziologie der Universität München, Konradstr. 6, 80801 München.

1. Einleitung[2]

Im universitären Bereich ist der Personal-Computer (PC) zu dem wichtigsten Arbeitsinstrument geworden, ohne das wissenschaftliches Arbeiten kaum mehr vorstellbar ist. MS-DOS war dabei von der Geburtsstunde des PC bis heute das Standard-Betriebssystem, mit dem eine ganze Generation von Geistes- und Sozialwissenschaftlern aufgewachsen ist. Seit einigen Jahren sind nun mehrere PC-Prozessorgenerationen auf dem Markt, die erheblich mehr Möglichkeiten bieten als das technisch beschränkte DOS erreichen kann. Mit dem Aufkommen dieser 32-Bit-Prozessoren ist der Einsatz leistungsfähiger Betriebssysteme auf preisgünstiger Massenhardware realisierbar. Für Wissenschaftler aus allen Bereichen ergibt sich dadurch ein potentieller Zugang zu neuen Verfahren, die computergestütztes Arbeiten wesentlich erleichtern können. Der Artikel soll die elementarsten Formen demonstrieren. Zunächst werden jedoch die zentralen Merkmale und Vertreter zeitgemäßer Alternativen zum veralteten MS-DOS vorgestellt.

2. Eigenschaften moderner PC-Betriebssysteme

Die aus Anwenderperspektive wichtigsten Charakteristika moderner PC-Betriebssysteme sind die Multitasking-Fähigkeit und die Möglichkeit, unter grafischen, fensterorientierten Oberflächen arbeiten zu können.

Multitasking bedeutet, daß mehrere Programme gleichzeitig und parallel abgearbeitet werden können. Dies ist dann z.B. sinnvoll, wenn eine Anwendung längere Zeit zur Ausführung braucht. Triviales Beispiel ist das Formatieren von Disketten. Ein nicht multitasking-fähiges Betriebssystem wie DOS blockiert in dieser Zeit die ganze Computereinheit. Ein multitasking-fähiges System hingegen ermöglicht, das zeitaufwendige Programm "in den Hintergrund zu legen". Die Hintergrundsitzung wird für den Benutzer unsichtbar abgearbeitet, während er in der Vordergrandsitzung mit dem gleichen oder einem anderen Programm weiterarbeiten kann.

Die zweite Neuerung, die mehr im Mittelpunkt dieses Artikels steht, ist die Ablösung von zeichenorientierten durch **grafische, fensterorientierte Oberflächen.** Mit "Oberfläche" ist hierbei das Erscheinungsbild von Programmen am Bildschirm gemeint. Grafische Benutzeroberflächen oder "Schnittstellen" werden im Fachjargon als GUIs (Graphical User Interface) bezeichnet. Vordergründigstes Ziel dieser GUIs ist es, den Anwender in seiner gewohnten Welt denken und arbeiten zu lassen, ohne ihn mit der Komplexität einer "anderen Welt" (Betriebssystem-Kommandos o.ä.) zu konfrontieren *(**Winograd und Flores*** 1992,

---

2 Für Verbesserungsvorschläge und Korrekturen zum Text möchte ich mich bei Mitarbeitern des Zentralarchivs bedanken.

S.269-270). Mittelpunkt eines GUIs ist der elektronische Schreibtisch oder "Desktop", der dem Benutzer seine gewohnten Organisationsmittel in grafischer Darstellung und in unterschiedlichen Bildschirmausschnitten (Fenstern) zeigt. In dieser Ansicht werden beispielsweise Bürowerkzeuge wie Ordner, Papierkorb oder Reißwolf als kleine Symbole (Icons) repräsentiert. Fortgeschrittene GUIs erlauben es, reale Büroprozesse analog und intuitiv auf dem elektronischen Desktop nachzuahmen. Soll z.B. ein Dokument vernichtet werden, nimmt man es im realen Büroprozeß in die Hand und gibt es in den Reißwolf. Auf dem Computer-Desktop wird der Vorgang völlig analog mit Maus und "Drag-and-Drop"-Technik nachgeahmt: das Dokumenten-Symbol wird auf das Reißwolf-Symbol gezogen ("drag") und fallengelassen ("drop"). Der Benutzer muß sich nicht darum kümmern, was sich hinter dem Bewegen der Icons verbirgt, sondern er kann sich auf die ihn eigentlich interessierenden Vorgänge konzentrieren.

Überlegt man sich den Umstieg auf solch grafisch ausgerichtete Betriebssysteme, so darf man sich als Anwender von den neuen Möglichkeiten nicht blenden lassen. Das entscheidende Kriterium für die Wahl eines modernen Betriebssystems ist der *Schutz* von bereits getätigten Softwareinvestitionen. Dies bedeutet für DOS-Benutzer, daß alte DOS-Software auch unter dem neuen Betriebssystem möglichst problemlos ablaufen soll. So gut die neue Plattform auch sein mag - man will schließlich auch noch mit der alten Software weiterarbeiten können und nicht auf diese verzichten müssen, nur weil man in einer anderen Umgebung arbeitet. Hierzu muß man wissen, daß das Ablaufenlassen von Programmen auf fremden Betriebssystemen per se nicht gewährleistet ist. Programme sind nämlich immer für bestimmte Systemplattformen entwickelt und ein Programm für Betriebssystem X läuft nicht auf Betriebssystem Y. Von einem anwenderfreundlichen, Softwareinvestitionen schützenden PC-Betriebssystem erwartet man zumindest, daß auf ihm auch noch alte DOS-Programme in Form von Emulationen[3] o.ä. einsetzbar sind.

3. UNIX, OS/2 und Windows

Die GUI-Idee des einfach handhabbaren Computers geht ursprünglich auf ***Allen Kay*** vom Xerox PARC (Palo Alto Research Center) zurück. ***Kay*** schuf in den siebziger Jahren das Konzept der grafischen Benutzeroberfläche und die Firma Apple setzte es Anfang der achtziger Jahre erstmals kommerziell um. Nach einem finanziellen Flop gelang Apple mit dem Macintosh der große Wurf. Die Oberfläche des Apple Macintosh gilt heute noch vielen als vorbildlich und nachahmenswert. Der "Mac" hat in Deutschland leider nie recht den Massenmarkt erobern und das Flair des "Exotischen" ablegen können, so daß wir hier nicht weiter darauf eingehen. An manchen Universitätsinstituten ist der "Mac" jedoch "Computer der Wahl".

3 Salopp formuliert, gaukelt eine Emulation einem Programm einfach eine bestimmte Betriebssystemumgebung vor.

Als die drei Hauptvertreter am Markt grafisch orientierter Betriebssysteme gelten UNIX, OS/2 und - mit technischen Abstrichen - Windows. Multitasking und GUIs waren in der Vergangenheit primär eine Domäne von teuren UNIX-basierten Rechnern wie Workstations. Leistungsfähige Workstations[4] werden vor allem in den naturwissenschaftlichen und mathematischen Fächern verwendet, wo i.a. größere Rechenpower benötigt wird als in den Sozial- und Geisteswissenschaften. UNIX-Varianten sind inzwischen aber auch für die populäreren PCs erhältlich. Diese UNIX-PC-Versionen sind von allen drei Hauptvertretern derzeit tatsächlich auch die leistungsfähigsten, sie haben aber unter anderem drei entscheidende Nachteile: erstens gibt es eine Vielzahl von UNIX-Dialekten (z.B. SCO-, Interactive-UNIX, UNIXware) mit unterschiedlichen, auf dem X-Window-System aufsetzenden grafischen Oberflächen (z.B. Motif, Open Look, PC-Solaris), zweitens fehlen die billigen, verbreiteten Massenprogramme, wie sie aus der DOS- und Windows-Welt vertraut sind und drittens ist uns kein UNIX-System bekannt, das eine befriedigende Emulationsmöglichkeit für DOS-Software bietet. In Bereichen, in welchen andere Kriterien als High-End-Rechenleistung wichtig sind, fristet UNIX deshalb bislang eher ein Schattendasein, wenngleich eine aufsteigende Tendenz erkennbar ist.

Speziell für die weitverbreiteten Familien der 386-, 486- und Pentium-(586-)[5] PC-Prozessoren sind OS/2 und Windows die derzeit beliebtesten Systemplattformen[6]. Das von IBM entwickelte OS/2 hat dabei die modernere und leistungsfähigere Architektur: es ist ein echtes 32-Bit-System und nutzt die Möglichkeiten der 32-Bit-Prozessoren (ab Intel 386) voll aus. Der Nachteil von OS/2 ist, daß Anwendersoftware, welche die *32-Bit-Fähigkeiten* von OS/2 verwertet, derzeit auf dem Softwaremarkt noch kaum verfügbar ist[7]. OS/2 hat aber im Gegensatz zu UNK den entscheidenden Vorteil, daß unter OS/2 als Basissystem sowohl DOS- als auch Windows-Programme parallel ausführbar sind. DOS- und Windows-Programme laufen unter OS/2 z.T. sogar schneller und absturzsicherer ab als in ihren "natürlichen" Umgebungen. OS/2 stellt somit eine Integrationsplattform dar, die dem Anwender die Möglichkeit gibt, seine bisher gewohnte Umgebung ohne jedwede Einschränkung nahtlos mit einer hochmodernen Betriebssystemarchitektur zu verbinden *(Glas et al.* 1992, S. 118). Damit ist der sanfte Übergang von der alten DOS- in die neue 32-Bit-Welt bestens gewährleistet.

4 Die Frage, wie man PCs von Workstations abgrenzt, ist nicht einfach zu beantworten, da die Grenzen immer mehr verschwimmen. Das klarste Unterscheidungsmerkmal liegt noch darin, daß Workstations grundsätzlich einen anderen Prozessortyp verwenden als PCs, nämlich sog. RISC-Prozessoren (statt CISC-Prozessoren bei PCs). Die anderen charakteristischen Merkmale von Workstations wie hochauflösende Grafikbildschirme oder UNIX als Betriebssystem können nicht mehr als Abgrenzungskriterium dienen, da sie sich vermehrt auch bei PCs finden.

5 Die PC-Prozessoren stammen ursprünglich ausnahmslos von der Firma Intel. Erst in neuerer Zeit gibt es billigere 386er- und 486er-Nachbauten z.B. von AMD oder Cyrix. Um einen besseren rechtlichen Schutz vor solchen Clones zu gewährleisten, wählte Intel als Namen für die neueste Prozessorgeneration nicht wie erwartet 586, sondern Pentium.

6 Die zur Zeit der Entstehung dieses Artikels aktuellen Versionen sind OS/2 2.1 und Windows 3.1.

7 Es gibt eine SPSS-Version für OS/2, diese wurde aber noch für das alte 16-Bit OS/2 1.x programmiert und nutzt damit die 32-Bit-Architektur der Version 2.x nicht aus.

Die grafische PC-Oberfläche mit der derzeit breitesten Akzeptanz - die "GUI fürs Volk" - ist aber Windows von Microsoft. Windows hat den Vorteil, daß es eine Vielzahl von billigen, populären Anwenderprogrammen gibt, die unter diesem Betriebssystem laufen (und damit, wie gesagt, auch unter OS/2). Ferner ist unter Windows die sanfte Migration gewährleistet, da DOS-Programme in Windows-Fenstern ausführbar sind. Der große Nachteil von Windows ist, daß es derzeit nur ein "Aufsatz" auf MS-DOS ist und dadurch erheblichen Limitierungen unterworfen ist [8] (z.B. kein leistungsfähiges Multitasking, vgl. unten). Technisch ist Windows im Vergleich zu UNIX oder OS/2 eindeutig die schlechtere Lösung. Microsoft bringt als Alternative und Anschluß an die Konkurrenz in diesen Tagen das High-End-System Windows NT (New Technology) auf den Markt. Dieses von DOS unabhängige 32-Bit-System soll die Leistungsstärke von UNIX erreichen, setzt allerdings erhebliche Hardware-Ressourcen voraus (z.B. mind. 486er Prozessor, 16 MB Hauptspeicher). Da diese de facto oft noch nicht gegeben sind, wird OS/2 oder die DOS-Variante von Windows für viele Benutzer in nächster Zeit das Betriebssystem ihrer Wahl bleiben [9].

Die Frage, ob Windows oder OS/2 bevorzugt werden sollte, ist in unserem Zusammenhang nicht entscheidend. Beide Systeme sind bezüglich der unten beschriebenen, elementaren Möglichkeiten ziemlich gleichwertig. Zwar wird in diesem Artikel Windows zugrundegelegt, die dargestellten Verfahren sind aber ziemlich analog auch unter OS/2 verwendbar.

Die Arbeitsmittel, welche die Desktops der einzelnen GUIs bereitstellen, variieren stark. Während Windows eine Ansicht verwendet, die sich eher an Applikationen und dem Dateisystem orientiert, arbeiten z.B. OS/2 und UNIXware davon unabhängig und erlauben, sich eine völlig eigene Hierarchie aus Dateien und Anwendungen zu schaffen [10]. Dies kann so gut gelingen, daß sich Chaoten auf ihrem Computer-Desktop ein ähnliches Bild bietet wie auf dem echten Schreibtisch. Die folgende Abbildung zeigt die applikations- und dateiorientierte Oberfläche von Windows. In dem rechten Fenster "Programm-Manager" sind drei weitere Fenster mit unterschiedlichen Typen von Anwenderprogrammen enthalten. Das linke Fenster "Datei-Manager" zeigt den Inhalt eines Verzeichnisses an. Die Icons A, B und C im Kopf des Datei-Manager-Fensters stellen bildlich Disketten- und Festplattenlauf-

8 Mit "Aufsatz" ist gemeint, daß Windows zum Betrieb DOS als Basissystem voraussetzt. Windows ist deshalb genaugenommen kein eigenständiges Betriebssystem. Manche sprechen nur von einer Oberfläche für DOS, aber das ist auch nicht richtig, da Windows eine andere Speicherverwaltung als DOS hat. Windows ist ein Zwitter zwischen Betriebssystem und reiner Oberfläche. Am passendsten erscheint wohl, von Windows als "Betriebssystem-Erweiterung" zu sprechen. Wir werden im folgenden aber darauf verzichten und einfachheitshalber weiter den Ausdruck "Betriebssystem" verwenden.

9 Microsoft positioniert NT neuerdings nicht mehr als Einzelplatz-System, sondern als Server-Betriebssystem, also als System, das ein Computernetz bedient. Als eigentlicher Windows-Nachfolger gilt die nächste Version 4.0, die ebenfalls unabhängig von DOS sein wird, aber weniger Hardwareressourcen voraussetzt als NT.

10 Es gibt für GUIs bestimmte Richtlinien wie z.B. die, daß das Kurzzeitgedächtnis durch übersichtliche Menüs, einfachen Bildaufbau etc. nicht stark belastet werden sollte. Die Hersteller der verschiedenen GUI-Plattformen bieten Style-Guidelines für Software-Entwickler an, um eine möglichst konsistente Benutzerführung zu gewährleisten. Auch Normierungsgremien warten mit Richtlinien auf. Für nähere Details hierzu vgl. *Hüskes und Shahrbabaki* (1993).

werke dar und können durch einen einfachen Mausklick aktiviert werden. Die einzelnen Icons in den Programm-Manager-Fenstern und auf dem Desktop ganz unten repräsentieren Anwendungsprogramme.

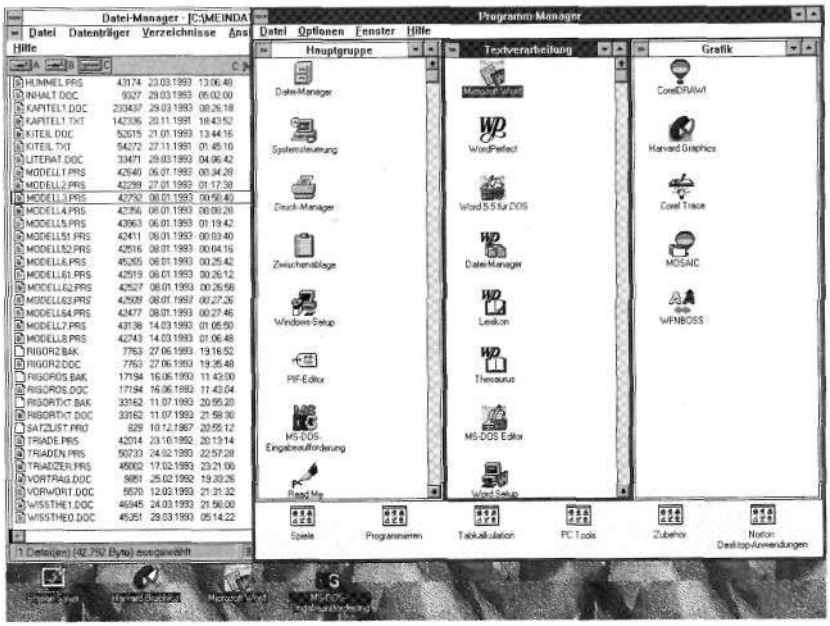

Abbildung 1: Der Windows-Desktop mit Datei- und Programm-Manager

4. Wer profitiert von grafischen Oberflächen?

Abgesehen von eingefleischten Hackern und Kommandozeilen-Freaks profitiert im Prinzip jeder von einer grafischen Oberfläche - eine entsprechende Hardwarekonfiguration vorausgesetzt[11]. Verglichen mit den alten und umständlichen Eingaben von DOS-Kommandos ist für den Durchschnitts-User jede GUI ein Bedienungsfortschritt. Dies trifft aus meiner

11 *Minimale* Hardwarevoraussetzung für *vernünftiges* und *ergonomisches* Arbeiten mit Windows ist aus meiner Sicht ein 386er mit 33 MHz, 4 MB Arbeitsspeicher, einer möglichst großen, schnellen Festplatte und leistungsfähigen Grafikkarte. Der Farbbildschirm sollte strahlungsarm sein (nach der Schwedennorm MPR-II oder TCO-92), wenigstens 15 Zoll Bildschirmdiagonale messen und 800 x 600 Punkte Auflösung darstellen können bei wenigstens 70 Hz augenschonender Bildwiederholrate. *Optimal* wäre ein 486er (oder gar Pentium) mit 66 MHz, 8 -16 MB Speicher und einem 17 Zoll Schirm mit flimmerfreier Auflösung bei 1024 x 768 Punkten. Insbesondere wenn man mit mehreren Programmen gleichzeitig arbeiten will, sollte an Hauptspeicher nicht gespart werden.

Erfahrung insbesondere für manche Nutzer aus den Geistes- und Sozialwissenschaften zu, die oft eine Abneigung gegen alles "Formale" haben und mit Bildern und Icons eher zurechtkommen. Allerdings ist der Desktop der Volks-GUI Windows nicht so konsequent realisiert und schön zu bedienen wie der des Apple Macintosh oder auch der von OS/2. Die Oberfläche ist auch nicht so einfach angelegt, daß man das System völlig intuitiv und ohne Einarbeitung handhaben kann. Dennoch ist ein grundlegender Fortschritt und eine Bedienungserleichterung zu erkennen.

Auf der Ebene der Anwenderprogramme ergibt sich grundsätzlich ein dreifacher Vorteil. Erstens sind Applikationen für GUIs i.a. leistungsfähiger und bieten vielfältigere Möglichkeiten als die entsprechenden DOS-Pendants. Zweitens sind die Programme einfacher und intuitiver zu bedienen, so daß der Lernaufwand minimiert wird und Gedächtniskapazität für andere Dinge frei wird als das Lernen von Befehlssequenzen. Drittens schließlich sind die Anwendungen in gewisser Weise normiert und funktionieren bei gleichen Aufgaben sehr ähnlich. Kennt man also ein Anwendungsprogramm, ist ein neues Programm schnell erlernbar.

Uns interessieren aber hier nicht die allgemeinen Vorzüge von GUIs, Windows oder Anwendungsprogrammen. Im Mittelpunkt stehen vielmehr die durch die neuen Betriebssysteme bedingten erweiterten wissenschaftlichen Arbeitsmöglichkeiten. Hier ergibt sich eine Nutzendifferenzierung, die man am besten an zwei typischen Benutzerprofilen unterscheiden kann.

Die Mitglieder der ersten Benutzergruppe können dadurch charakterisiert werden, daß sich ihre wissenschaftliche Arbeitstätigkeit am Computer ausschließlich auf die Produktion von *Dokumenten* umgangssprachlicher oder auch formalerer Natur beschränkt (Zeitschriftenartikel, Bücher, Unterrichtsunterlagen etc.). Der Computer wird von dieser Gruppe weitgehend als reine Textverarbeitungsmaschine gesehen und benutzt und der Desktop wird allenfalls noch für Verwaltungsarbeiten verwendet. Der Prototyp ist der "rein geisteswissenschaftlich" arbeitende, ausschließlich Fließtext erzeugende Benutzer (im Gegensatz zu Nicht-Fließtextproduzenten, die mathematische Symbole oder Formeln in ihren Texten verwenden). Reine Fließtextproduzenten haben am wenigsten von GUIs. Die Vorteile sind im Prinzip beschränkt auf die intuitivere Bedienbarkeit der Maschine und WYSIWYG bei der Textverarbeitung. Mit **WYSIWYG** (what you see is what you get) ist gemeint, daß der Text so auf dem Bildschirm dargestellt wird, wie er später auf dem Printer ausgedruckt wird[12]. Dies ist bei den herkömmlichen zeichenorientierten Anwendungen in der Regel nicht der Fall.

12 Konkret bedeutet dies z.B., daß Textteile in Kursiv- oder Fettschrift auf dem Bildschirm ebenfalls kursiv oder fett dargestellt werden. Die WYSIWYG-Darstellung vermittelt anders als die zeichenorientierte Ansicht einen genauen Eindruck davon, wie die bedruckte Seite später aussehen wird.

Für die zweite Benutzergruppe sind GUIs schon wesentlich interessanter und bieten substantiell mehr als bloße Druckbilddarstellung oder Bedienungsvereinfachung. Die wissenschaftlich essentielle Benutzung des Computers ist hier nicht auf reine Textproduktion reduziert. Vielmehr wird der Rechner als bevorzugtes Werkzeug für unterschiedlichste wissenschaftliche Arbeitstätigkeiten angesehen, von welchen Textproduktion nur ein kleiner Teil ist. Beispielsweise ist eine empirische Untersuchung statistisch auszuwerten und die Resultate sind tabellarisch und grafisch darzustellen. Ergebnisse oder Teile davon müssen in Publikationen in Form von Objekten wie Grafiken oder Tabellen eingebunden werden. Für bestimmte, spezielle Problemstellungen sind eigene Programme zu entwickeln und Programmoutputs sind in das Dokument zu integrieren. Diese Gruppe von Benutzern setzt sich aus formal und empirisch orientierten Wissenschaftlern unterschiedlichster Fächer zusammen, die mit verschiedenartigsten Werkzeugen und Methoden arbeiten.

Aus dem eben gesagten läßt sich die Hypothese ableiten, daß GUIs und darauf basierende Techniken im wissenschaftlichen Bereich umso nützlicher werden, je gestreuter die zu bearbeitenden Aufgabenstellungen sind und je weiter man sich von reiner Fließtextproduktion entfernt. Diese Behauptung soll nun anhand eines Dissertationsprojektes verdeutlicht werden.

Wir haben die Windows-Umgebung verwendet um wissensbasierte Computermodelle zu entwickeln. Diese sollten sozialwissenschaftliche Gleichgewichtstheorien repräsentieren und u.a. für Experimentierzwecke benutzt werden. Die Modelle wurden in der Programmiersprache Prolog implementiert, der verwendete Interpreter war IF/Prolog für DOS. Ohne weiter auf inhaltliche Fragestellungen einzugehen, sei nur angemerkt, daß es bei den Theorien in erster Linie um die Behandlung graphentheoretischer Konzepte ging, die bestimmte kognitive und soziale Strukturen modellierten. Die in diesem Projekt zu verrichtenden Arbeiten lassen sich kurz wie folgt beschreiben. Neben der Erstellung des eigentlichen Fließtextes gab es breite Textteile mit logischen und mengensprachlichen Symbolen sowie mathematische Formeln, die in den Text zu integrieren waren. Parallel zur Texterfassung wurden die Prolog-Programme geschrieben. Sowohl Prolog-Code als auch Programmoutput mußten in das Dokument übernommen werden. Weiter mußten Grafiken, in der Hauptsache mathematische Graphen, gezeichnet und in den Text integriert werden. Das Arbeiten in dieser heterogenen Aufgabenstellung implizierte, daß Informationen aus unterschiedlichen Anwendungsprogrammen möglichst

- gleichzeitig am Bildschirm darstellbar und einsehbar sowie
- problemlos und schnell zwischen Applikationen transferierbar sein mußten.

Auf DOS-Ebene mit DOS-Programmen ist ein solches Arbeiten unmöglich, GUIs mit entsprechenden Anwendungsprogrammen bieten hingegen ein Optimum an Unterstützung,

Komfort und Leistung. Die hier verwendeten Applikationen waren die Windows-Versionen von Word, Harvard Graphics, Norton Editor und die DOS-Version von Prolog.

5. Scientific Word Processing

Betrachten wir zunächst die reine Textproduktion wissenschaftlicher Dokumente mit grafisch orientierten Textverarbeitungsprogrammen. Zum Erfassen wissenschaftlicher Dokumente reichen im Normalfall - wenn sie nicht extrem mathematisch ausgerichtet sind - verbreitete und preiswerte Standard-Windows-Programme wie Word oder Wordperfect völlig aus. Diese bieten im Vergleich zu zeichenorientierten Programmen eine erhebliche Bedienungsvereinfachung. Kleine Symbole simulieren z.B. Schalter, mit denen bestimmte Aufgaben wie Speichern, Drucken oder Textformatierung durch einen einzigen Mausklick ausgeführt werden können. Abb. 2 zeigt diese Symbole, die sich der Benutzer selbst nach Bedarf zusammenstellen kann, im Kopf des Programmfensters.

Von den üblichen Textverarbeitungswerkzeugen wurden bei der Texterstellung insbesondere Druckformatvorlagen, der Thesaurus (Synonymlexikon), die Fußnotenverwaltung, die Sortierfunktion (z.B. Sortieren des Literaturverzeichnisses), das automatische Anlegen von Inhaltsverzeichnissen oder die Erstellung eines Index verwendet. Für einen längeren Text im Umfang von 200-300 Seiten empfiehlt es sich, diesen in mehrere Dateien aufzuteilen. Sehr nützlich ist dabei, daß man mehrere Textdateien gleichzeitig in verschiedenen Fenstern darstellen oder im Hintergrund offen halten, bequem mit einem Mausklick von Dokument zu Dokument wechseln oder Textteile mühelos austauschen kann.

Eine neuere Technik, die beim Textentwurf nicht mehr wegzudenken ist, ist das bereits oben angedeutete "Drag and Drop". Mit diesem Verfahren können Textbereiche einfach markiert und mit der Maus verschoben werden. Ganz praktisch ist auch das leichte Wechseln zwischen verschiedenen Darstellungsarten wie Druckbild- oder Konzeptdarstellung und die Möglichkeit, das Dokument auf verschiedene Größen zu "zoomen".

All diese Mittel werden im Prinzip auch in nicht-wissenschaftlichen Dokumenten verwendet und sollen hier nicht weiter behandelt werden. Das Erfassen wissenschaftlichen Fließtextes am Computer, einige Richtlinien, die dabei zu beachten sind und die o.g. Funktionen sind ausführlicher dargestellt in ***Nitze*** (1993). Dort finden sich auch weitere Literaturhinweise zum Verarbeiten von wissenschaftlichen Texten.

Mir ist hier ein anderer Aspekt wichtiger. Eine zentrale Stellung in diesem Dissertationsprojekt und vielen anderen wissenschaftlichen Dokumenten nehmen Sonderzeichen ein, und zwar insbesondere mathematische Symbole und Formeln. Word stellt ebenso wie

Wordperfect einen Zeichensatz mit mathematischen Symbolen zur Verfügung, der für den Normalgebrauch in der Regel ausreichen dürfte. In Word kann auf Symbole dieses Zeichensatzes durch Wahl des Pulldown-Menüs *Einfügen* in der Menüleiste zugegriffen werden (siehe Abb. 2, die Menüleiste ist die Zeile, die mit dem Namen "Datei" beginnt). Aus dem - in Abb. 2 nicht sichtbaren - Menü ist dann der Befehl *Sonderzeichen* zu wählen, worauf das Sonderzeichen-Fenster wie in Abb. 2 zur Verfügung steht. Das benötigte Symbol kann nun durch Doppelklick ausgewählt werden und wird in den Text an Cursorposition integriert.

Werden die Sonderzeichen oft benötigt, so empfiehlt sich, diese Funktion als kleines Symbol auf die Funktionsleiste zu legen (die Funktionsleiste ist die Zeile mit den Symbolen, welche rechts mit dem Lupen-Symbol endet). Das Definieren neuer Symbole für die Funktionsleiste ist im Menü *Extras - Einstellungen* vorzunehmen. In dem Bildschirmausschnitt verbirgt sich die Sonderzeichenfunktion auf der Funktionsleiste hinter dem siebten Icon von links.

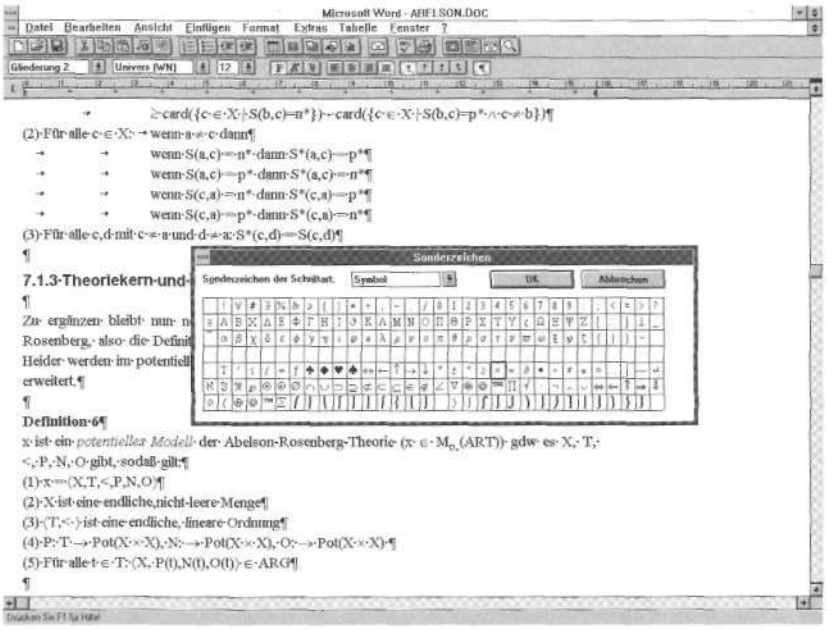

Abbildung 2: Die Windows-Textverarbeitung Word mit aktiviertem Sonderzeichen-Fenster

Sind viele Sonderzeichen einzufügen, wird das Arbeiten in Word etwas mühsam. Nach jedem Doppelklick auf ein Symbol des Zeichenfensters wird das Fenster nämlich automatisch geschlossen und muß für jedes neu einzufügende Symbol erst wieder geöffnet

werden. Bei Zeichen vom gleichen Typ kann man sich mit der Wiederholfunktion (F4) behelfen, wie etwa in dem folgenden Ausdruck mit den Sonderzeichen ⟨, ⟩ und ∨:

$\langle\langle x_0,x_1\rangle \vee \langle x_1,x_0\rangle, \langle x_1,x_2\rangle \vee \langle x_2,x_1\rangle, \ldots, \langle x_n,x_0\rangle \vee \langle x_0,x_n\rangle\rangle$

In diesem Fall kann nach einmaliger Aktivierung des Sonderzeichenfensters und Einfügen des entsprechenden Symbols derselbe Symboltyp mit der Wiederholfunktion beliebig oft eingesetzt werden ohne wiederholte Öffnung des Sonderzeichenfensters. Für diesen Ausdruck ist das Sonderzeichenfenster somit nur dreimal zum Einfügen der drei Zeichentypen zu aktivieren. Die Symbole sind dann nur noch mit Drag-and-Drop entsprechend anzuordnen. Dennoch ist die Integration vieler unterschiedlicher Symbole durch ständiges Fensteröffnen nicht sehr benutzerfreundlich und sollte in der nächsten Version geändert werden.

In Wordperfect ist das Einsetzen von Sonderzeichen benutzerfreundlicher gelöst. Im Gegensatz zu Word bietet das Zeichensatzfenster von Wordperfect nämlich die Option, das Fenster beliebig lange offen zu halten. Es muß somit nicht jedesmal neu gestartet werden und der Zugriff auf die Symbole des Zeichensatzes kann viel schneller erfolgen.

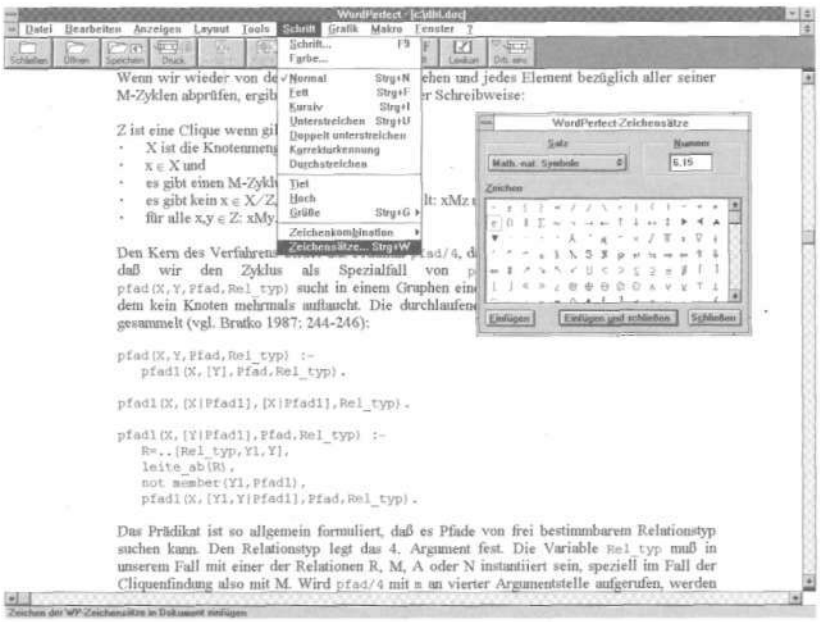

Abbildung 3: Die Windows-Textverarbeitung WordPerfect mit aktiviertem Zeichensatz-Fenster für mathematisch-naturwissenschaftliche Symbole

Komplexere Formeln mit Wurzeln, Brüchen etc. sind durch einfaches Einfügen mathematischer Symbole natürlich nicht zu erzeugen. Für komplexere Ausdrücke stellt Word einen Formel-Editor zur Verfügung, eine Spezialversion des MathType-Editors, der relativ einfach und intuitiv zu bedienen ist. Der Formel-Editor wird unter *Einfügen - Objekte* aktiviert und die Formel wird unter Verwendung von Symbol- und Vorlagenpaletten generiert. Die fertig erstellte Formel kann über die Zwischenablage (siehe unten) in den Text eingebaut werden. Zur Änderung der Formel und dem Aufruf des Editors genügt ein Doppelklick auf das Formelobjekt im Text.

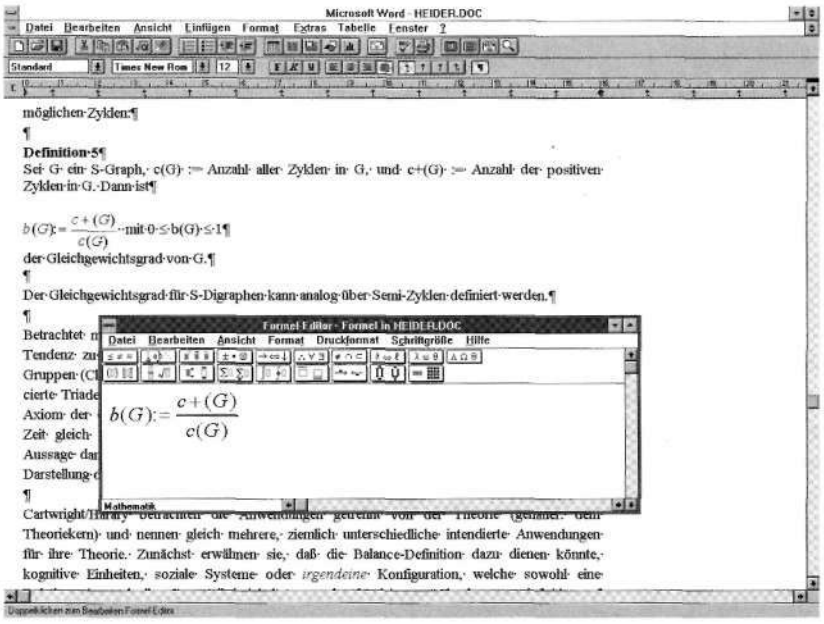

Abbildung 4: Der Formel-Editor in Word

Bislang wurden nur Optionen grafisch orientierter *Textverarbeitung* für das Erstellen wissenschaftlicher Dokumente betrachtet, nun soll auf den eigentlichen Vorteil von Windows oder allgemein GUIs eingegangen werden: das Zusammenspiel unterschiedlicher Informationsquellen und Programme.

6. Gleichzeitige Darstellung verschiedener Informationen

Der folgende Bildschirm illustriert an einem konkreten Beispiel den Vorteil der Darstellung unterschiedlicher Anwendungen in einzelnen Fenstern. Im Fenster auf der linken Seite wird ein in Harvard Graphics gezeichneter Graph dargestellt, der auf einem kleinen Datensatz beruht. Im Fenster rechts oben erscheint der Output des Prolog-Programms, das als Input den links repräsentierten Datensatz bearbeitet. Im Fenster rechts unten ist schließlich der Programmcode im Editor zu sehen. Hierzu sei erwähnt, daß der Prolog-Interpreter keinerlei Entwicklungsumgebung beinhaltet, so daß der Editor aus dem Norton Desktop zum Programmieren verwendet wurde. Ist die Fensterdarstellung zu klein, so kann durch einen Mausklick (bzw. ALT-RETURN bei DOS-Fenstern) auf Vollbilddarstellung umgeschaltet werden und wieder zurück zur ursprünglichen Fenstergröße.

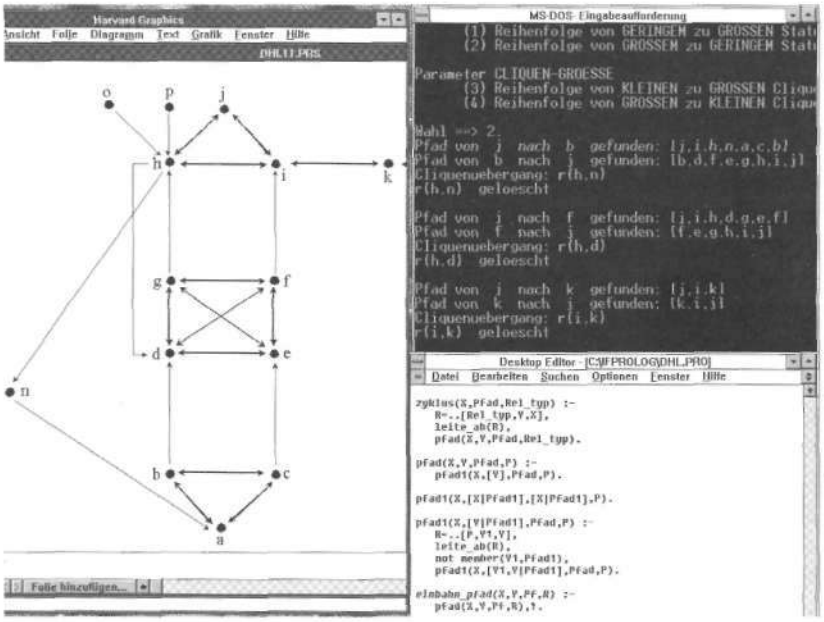

Abbildung 5: Simultane Darstellung unterschiedlicher Informationen in drei Fenstern

Der Vorteil des fensterorientierten wissenschaftlichen Arbeitens am PC ist anhand des Beispiels unmittelbar einsichtig. Durch die Möglichkeit, mehrere Informationsquellen parallel auf dem Schirm darzustellen, kann der Lösungsweg anhand der Grafik und des Programmoutputs leicht verfolgt und die Korrektheit des Programms überprüft werden. Korrekturen

des Programmcodes können unmittelbar im Editorfenster erfolgen. In dem vorliegenden Beispiel sucht das Programm Pfade zwischen Cliquen und entfernt bestimmte Kanten (Pfeile) aus der Datenbasis, die von Cliquen zu anderen Cliquen führen. Cliquen sind hierbei graphentheoretisch als maximal vollständige Teilgraphen zu verstehen, z.B. sind in der Grafik links von Abb. 5 {i,h,j}, {i,k} oder {g,f,d,e} Cliquen. In dem Beispiel kann unmittelbar verifiziert werden, daß das Programm wie beabsichtigt die Kanten h-n, h-d und i-k entfernt.

Die Repräsentation unterschiedlicher Informationsquellen in Fenstern ist natürlich nicht nur bei Programmiertätigkeiten nützlich. Ein empirisch arbeitender Wissenschaftler kann analog z.B. Daten, Grafiken sowie Output und Anweisungen an ein Statistikprogramm in mehreren Fenstern darstellen und die jeweils benötigten Informationen unmittelbar oder mit einigen Mausklicks in Sekundenbruchteilen einsehen. Durch die fensterorientierte Informationsdarstellung ist in jedem Fall ein komfortableres und schnelleres Arbeiten möglich als mit reinen Vollbilddarstellungen.

7. Datenaustausch zwischen Windows-Programmen

Wie in fast jede wissenschaftliche Publikation mußten auch in den Dissertationstext Objekte wie Grafiken, Tabellen oder Ausgaben anderer Programme integriert werden. Das Programm, mit dem ein Objekt erzeugt wurde, ist dabei die Quellanwendung (z.B. eine Grafik mit Harvard Graphics), das Programm, in welches dieses Objekt übernommen werden soll, die Zielanwendung (z.B. das Textsystem). Die Übernahme von Objekten einer Quellanwendung in eine Zielanwendung ist unter Windows besonders einfach, so daß man sich von den Zeiten des Einklebens endgültig verabschieden kann. Allerdings muß beim Datentransfer zwischen Windows- und Non-Windows-(DOS-) Applikationen unterschieden werden. Beschränken wir uns vorerst auf den Datenaustausch zwischen reinen Windows-Programmen, so gibt es im wesentlichen drei Transfermöglichkeiten:

1. Datenaustausch über die Zwischenablage
2. Einbetten von Objekten
3. Verknüpfen von Objekten.

Die 1. Möglichkeit ist der einfachste Weg: das zu transferierende Objekt wird über die Zwischenablage (Clipboard) vom Quellprogramm ins Zielprogramm kopiert. Das Clipboard dient hierbei als temporärer Zwischenspeicher. Im einzelnen sind folgende Schritte auszuführen:

1. das Objekt wird in der Ursprungs- oder Quelldatei *markiert,*
2. **es** wird im Quellprogramm über das Pulldown-Menü *Bearbeiten - Kopieren* in die Zwischenablage *kopiert,*

3. die Zielanwendung wird geöffnet und
4. das Objekt wird über *Bearbeiten - Einfügen* in das Zieldokument *eingefügt.*

Die Arbeitsschritte zum Datentransfer über das Clipboard sind in allen Windows-Programmen gleich. Dieser quasi normierte Austausch zwischen Windows-Applikationen ist ein weiterer, nicht zu unterschätzender Vorteil, wenn man bedenkt, daß fast jedes DOS-Programm eine eigene Regelung für Datenimport und -export hat.

Mit dem einfachen Austauschverfahren kann beispielsweise ohne viel Aufwand der Programmcode in das Textdokument transferiert werden. In der folgenden Bildschirmdarstellung wird das Ursprungsprogramm - der Editor - im rechten Fenster und das Zielprogramm - Word - im linken, großen Fenster gezeigt. Der markierte, schwarz unterlegte Codeabschnitt im Norton Editor ist *über Bearbeiten - Kopieren* in die Zwischenablage (kleines Fenster) gelegt worden und kann nun in die Word-Zieldatei über *Bearbeiten - Einfügen* integriert werden.

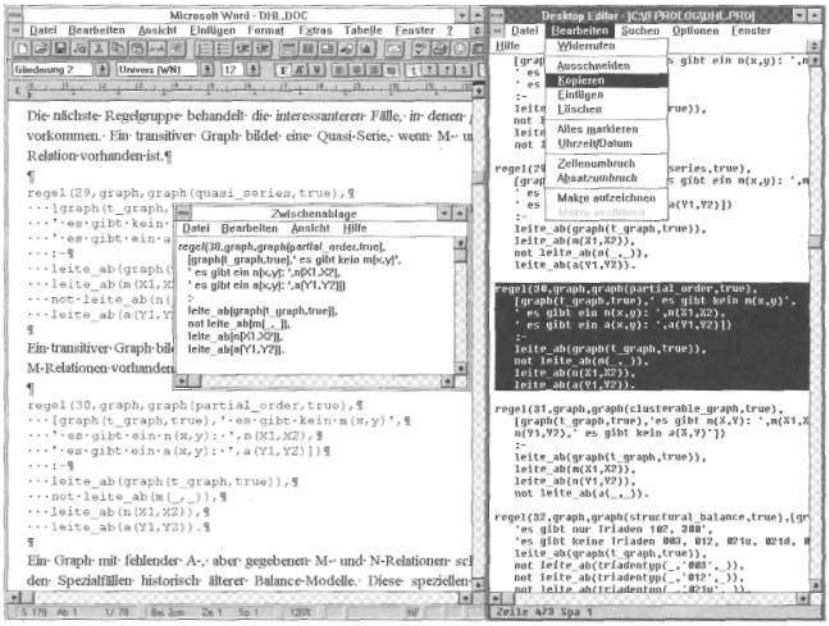

Abbildung 6: Datentransfer vom Editor (rechts) zur Textverarbeitung (links) über die Zwischenablage (kleines Fenster Mitte)

Die Methode des beschriebenen Datenaustausches über die Zwischenablage ist sehr einfach und grundsätzlich bei allen Windows-Programmen anwendbar. Der Nachteil dieses Verfahrens ist jedoch, daß der Transfer statisch ist und keine Verbindung zwischen Originaldaten und Kopie besteht. Wird der dunkel unterlegte Programmteil beispielsweise geändert, dann stimmt dieser nicht mehr mit dem Code im Textdokument überein und muß erneut aktualisiert werden. Bei Dokumenten, die immer wieder auf den neuesten Stand gebracht werden müssen, kann dies auf Dauer lästig sein.

Als Alternative zum statischen Datenaustausch über das Clipboard bietet sich das dynamische OLE-Verfahren an [13]. **OLE** steht für **"Object Linking and Embedding"** und wird im Deutschen mit "Verknüpfen und Einbetten" (von Objekten) wiedergegeben. OLE erweitert die Transfermöglichkeiten, ist aber erst ab Windows Version 3.1 und mit Anwendungsprogrammen möglich, weiche dieses neue Transferprotokoll unterstützen. Bei dem OLE-Verfahren wird das Programm, in das die Daten eingefügt werden, als Client bezeichnet (oben: Word), die Applikation, welche die Daten zur Verfügung stellt, als Server (oben: Norton Editor)[14].

Zwischen OLE und dem alten Transfer über die Zwischenablage gibt es zwei wesentliche Unterschiede. Ein erster Unterschied ist, daß im Gegensatz zum Austausch über das Clipboard die Clientanwendung Daten einlesen kann, ohne daß sie deren Format kennen muß. Hierzu gehören z.B. auch Ton- und Videosequenzen. Der zweite, wichtigere Unterschied liegt darin, daß bei OLE die Verbindung des eingefügten Objekts zum Quellprogramm erhalten bleibt, während diese bei Einfügen über das Clipboard verloren ist. In dem obigen Word-Dokument ist beispielweise der Applikation Word nicht bekannt, von welcher Quelle das eingefügte Textobjekt stammt. Bei einem über OLE eingefügten Objekt existiert hingegen ein Verweis auf die Quelle und ein Doppelklick auf das Objekt im Client Word würde die Serverapplikation Norton Editor starten.

Grundsätzlich stellt OLE zwei Verfahren zur Verfügung, Linking (Verknüpfen) und Embedding (Einbetten). **Linking** bedeutet, daß das Objekt fest mit dem Original verknüpft ist und jede Änderung des Originals automatisch im Client übernommen wird. Das Objekt wird dabei nur einmal abgespeichert und nicht (mehrmals) zusammen mit den Dokumenten. Die Dokumente enthalten lediglich einen Verweis auf das Objekt. Im Fall der Verwendung des gleichen Objekts in mehreren Dokumenten spart Linking also Speicherplatz, denn die Quellinformation ist nur einmal vorhanden. Linking bietet sich sinnvollerweise an, wenn eine Information in verschiedenen Dokumenten verwendet werden soll, da eine Änderung im Original automatisch Änderungen in den Clients zur Folge hat.

13 Vgl. zu OLE insbesondere ***Kühn von Burgsdorff*** (1992).

14 Es sei nur angedeutet, daß es noch ein weiteres Transferprotokoll gibt: DDE oder Dynamischer Datenaustausch. DDE ist aber weniger leistungsfähig als OLE und nur nach langem Handbuchstudium sinnvoll anzuwenden. Es soll deshalb hier nicht weiter behandelt werden.

Embedding arbeitet hingegen nicht mit dem Original, sondern mit einer *Kopie* des Originals. Geändert wird bei Embedding also nur das Dokument, welches gerade bearbeitet wird und das Objekt wird zusammen mit dem Dokument abgespeichert. Dies entspricht weitgehend dem normalen Einfügen über die Zwischenablage. Der Unterschied ist lediglich, daß eine Referenz auf das Programm existiert, mit dem es generiert wurde. Einbetten ist dann sinnvoll, wenn Informationen nicht von verschiedenen Dokumenten geteilt werden sollen.

Bei beiden Verfahren läßt sich die Quellapplikation durch einen Doppelklick auf das Client-Objekt starten und sofort im Original bearbeiten. Die Auswirkung ist dann unterschiedlich: bei Änderungen in verknüpften Objekten werden die Modifikationen in alle Dokumente übertragen, die auf diese Information zugreifen. Bei Embedding hingegen ändert sich nur das Dokument, welches man gerade bearbeitet.

Im wissenschaftlichen Alltag dürfte Embedding die besseren Chancen haben als Linking, da es die Arbeit ohne zusätzlichen Verwaltungsaufwand erleichtert. Man kann sich vorstellen, daß das Arbeiten mit einem Netz von OLE-Links zumindest problematisch ist. Beispielsweise kann das Löschen oder Verschieben von Dateien, auf die sich andere Dokumente beziehen, fatale Folgen haben. Ebenso kann man nicht einfach Dokumente mit gelinkten Verbindungen auf einen anderen PC - z.B. einen Notebook - kopieren, da immer auch alle zugehörigen Datenquellen vorhanden sein müssen. Davon abgesehen ist Linking nicht ganz leicht zu verstehen und bedarf zumindest einer gewissen Lernphase und Einübung.

Harvard Graphics in der aktuellen Version 2.0 unterstützt OLE [15]. Der folgende Bildschirmausschnitt zeigt eine mit Harvard Graphics 2.0 erstellte und in das Word-Dokument *eingebettete* Grafik. Um die Grafik zu modifizieren genügt ein Doppelklick auf das Objekt im Word-Dokument: mit dem Doppelklick wird die Serverapplikation Harvard Graphics einschließlich der Grafik aufgerufen (rechtes Fenster). Dies ist der in Abb.7 dargestellte Zustand des Bildschirms. Die Grafik läßt sich jetzt im Grafikprogramm modifizieren. Die Änderungen werden nach Abschluß sofort im Word-Dokument übernommen.

8. Datenaustausch von DOS nach Windows

Bislang wurden nur Möglichkeiten des Datentransfers *innerhalb* von Windows-Programmen genannt. Von vielen Programmen existieren aber nur DOS-Versionen oder es sind nur solche verfügbar. Auch in diesen Fällen lohnt es sich, Windows als Betriebsumgebung zu

15 Harvard Graphics in der von mir verwendeten alten Version 1.01 unterstützte OLE leider nicht. Bei erforderlichen Änderungen der Grafik im Dokument mußte zunächst immer die alte Grafik umständlich im Word-Dokument gelöscht, Harvard und die entsprechenden Grafikdatei aktiviert, die Grafik modifiziert und erneut über die Zwischenablage in das Dokument eingefügt werden.

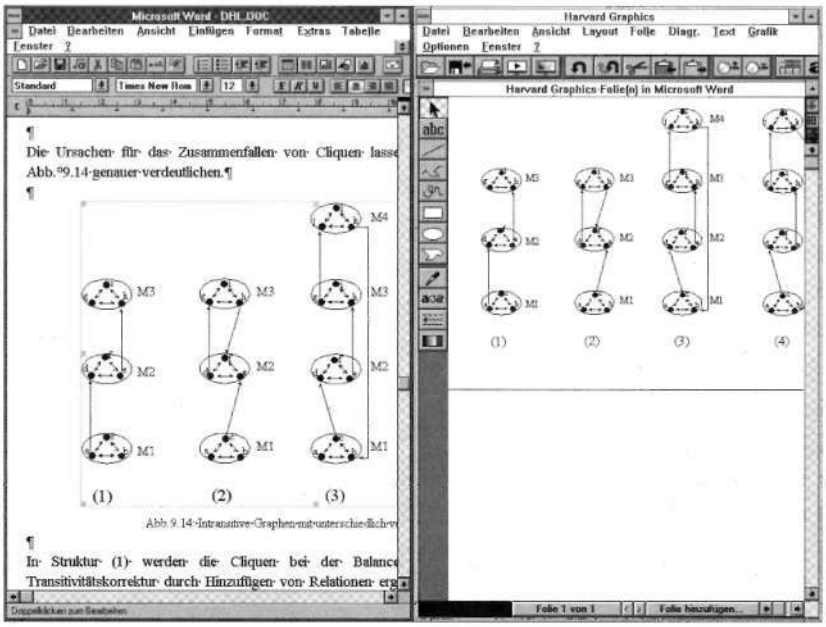

Abbildung 7: Datentransfer mit Embedding. Die Grafik wurde in das Textdokument eingebettet. Jeder Doppelklick auf das Grafikobjekt startet die Serverapplikation (hier: Harvard Graphics 2.0) mit dem Objekt

benutzen. Der Datenaustausch von einer DOS- in eine Windows-Applikation ist zwar weniger bequem, aber trotzdem im Vergleich zu reinen DOS-Anwendungen eine erhebliche Erleichterung. Als Transferverfahren verwendet man wieder die Zwischenablage.

Für das Projekt mußten insbesondere Ausgaben der Prolog-DOS-Programme in die Dokumente transferiert werden. Hierzu führt man am besten die DOS-Anwendung in einem Fenster aus und produziert die zu übernehmenden Programmoutputs in diesem Fenster. Die simpelste Methode besteht nun darin, Ausschnitte aus dem DOS-Bildschirm in die Zwischenablage zu übernehmen. Für die Übernahme ist zunächst im Systemmenü der DOS-Anwendung (linkes oberes Kästchen in der Menüleiste) *Bearbeiten - Markieren* zu aktivieren und der zu übernehmende Ausschnitt zu markieren. Anschließend wird (entweder durch einen Klick auf die rechte Maustaste oder wieder über das Systemmenü mit *Bearbeiten - Kopieren)* der markierte Ausschnitt in die Zwischenablage gelegt und kann nach Wechsel in die Zielanwendung über *Bearbeiten - Einfügen* ins Zieldokument eingesetzt werden.

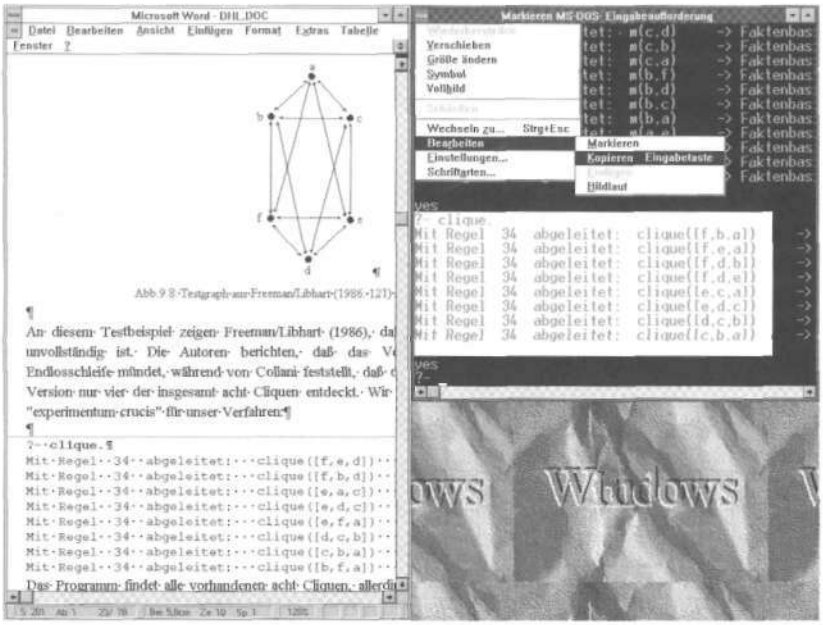

Abbildung 8: Datentransfer von einer DOS- in eine Windows-Anwendung

Die Übernahme von Programmoutput eines DOS-Programms wird allerdings aufwendiger, wenn die zu übernehmenden Ausgaben eine Fensterseite überschreiten. Der Grund ist, daß maximal eine volle Bildschirmseite markiert werden kann. In diesem Fall muß der Transferprozeß in mehrere Schritte aufgeteilt werden. In jedem Schritt wird solange gewartet, bis die DOS-Ausgabe eine neue Fensterseite füllt, die Ausgabe angehalten (z.B. mit Ctrl-S) und die Daten in das Zieldokument wie oben beschrieben transferiert. Nach jedem Datentransfer wird die Prozedur wiederholt, bis alle gewünschten Werte übertragen sind. Alternativ kann man natürlich die Programmausgabe in eine Datei umleiten und den Dateiinhalt in das Textdokument einfügen [16].

16 Es gibt noch weitere Möglichkeiten, Informationen aus DOS- und Windows-Programmen in Dokumenten zu verwenden. Hierzu gehört etwa die Momentaufnahme eines ganzen Bildschirms oder Bildschirmausschnitts in Form eines Bitmaps. Beispielsweise wird durch Betätigen der DRUCK-Taste der aktuelle Bildschirminhalt in die Zwischenablage kopiert und läßt sich in Dokumente einfügen. Auf diese Weise wurden die im Text verwenden Bildschirmkopien erzeugt. Eine systematische Zusammenstellung aller Verfahren findet sich Windows Benutzerhandbuch (Microsoft 1992).

Diese Möglichkeit des Datentransfers, die hier am Beispiel der Ausgaben eines Prolog-Programms demonstriert wurde, gilt natürlich für jedes DOS-Programm. Ein empirisch arbeitender Wissenschaftler, der z.B. seine Daten statistisch mit einem DOS-Programm wie SPSS auswertet, kann völlig analog die SPSS-Ausgaben und -Ergebnisse in das Textdokument integrieren. Durch solche fensterorientierte Arbeitstechniken kann der Substanzwissenschaftler nicht nur schneller arbeiten, es ist auch angenehmer und man kann sich besser auf seine inhaltliche Arbeit konzentrieren.

9. Multitasking

Wir wollen abschließend noch auf das eingangs erwähnte Multitasking eingehen. Multitasking hat weniger mit der Oberfläche zu tun, als mit den grundlegenden Eigenschaften eines Betriebssystems. Wie oben bereits erwähnt, ermöglicht Multitasking das gleichzeitige Arbeiten mit mehreren Programmen, was insbesondere dann sinnvoll ist, wenn ein Programm längere Zeit für Berechnungen braucht. Jeder empirische Forscher weiß aus persönlicher Erfahrung, wie zeitaufwendig manche Statistikauswertungen mit umfangreichen Datensätzen sein können. Rechenzeiten von mehreren Stunden sind bei großen Datenmengen bzw. komplexeren Statistikroutinen nicht selten. Ein anderes Beispiel sind Simulationsprogramme, bei denen noch extremere Zeiten auftreten können. Eine Computersimulation eines N-Personen-Gefangenendilemmas blockierte eine 286er-Maschine z.B. mehrere Tage ***(Diekmann und Manhart*** 1989). Im vorliegenden Fall kann das Prolog-Programm für die Auswertung umfangreicher Graphen abhängig von der Datenmenge ebenfalls sehr lang brauchen. In beiden Fällen stellt ein multitasking-fähiges System zweifellos eine Arbeitserleichterung dar, da der Benutzer während der Abarbeitung des Programms andere Aufgaben verrichten kann und das System nicht permanent blockiert ist.

Die Multitasking-Fähigkeiten von Windows sind im Vergleich zu anderen Systemen allerdings sehr beschränkt. Windows arbeitet mit kooperativem Multitasking im Gegensatz zu OS/2, das auf preemptivem Multitasking basiert. **Kooperatives Multitasking** setzt eine kooperative Zusammenarbeit von Applikationen voraus. Dies bedeutet, daß die Programme selbst die Prioritäten regeln und nicht das Betriebssystem. Da es viele Programme gibt, die sich selbst eine hohe Priorität und damit einen Großteil der gesamten Prozessorzeit zugestehen (und dies vom Benutzer nicht zu beeinflussen ist), "verhungern" andere Programme aufgrund fehlender Rechenzeit. Daneben besteht die Gefahr, daß ein Absturz eines Programmes auch den Absturz anderer Programme oder des ganzen Systems nach sich zieht. Beim **preemptiven Multitasking** hingegen laufen die Programm völlig unabhängig voneinander in einem geschützten Bereich ab und können sich nicht gegenseitig beeinflussen. Die Rechenzeit wird dynamisch verwaltet und Prozessorzeit grundsätzlich vom Betriebssystem zugewiesen. Will man also professionell mit mehreren Programmen gleichzeitig arbeiten, ist OS/2 (oder Windows NT) eindeutig die bessere Wahl. Ein Versuch von mir, unter

OS/2 die o.g. N-Personen-Simulation parallel in mehreren DOS-Fenstern ablaufen zu lassen, klappte hervorragend.

10. Schlußbemerkungen

Mit Ausnahme des letzten Punktes arbeiten alle hier beschriebenen Verfahren unter Windows und der Bedingung einer leistungsfähigen Rechnerkonfiguration problemlos und flott. In einer reinen DOS-Umgebung hätte sich das Projekt erheblich zeitaufwendiger und mühsamer gestaltet.

Unter Profis hat Windows den Ruf von mangelnder Stabilität, was ich nicht bestätigen kann. Während des ganzen Projekts gab es keine größeren Probleme oder Programmabstürze. Wer Windows mißtraut, auf Multitasking oder ein echtes 32-Bit-System Wert legt, sollte auf OS/2 umsteigen. Man kann nur allen Geistes- und Sozialwissenschaftlern mit heterogenen Aufgabenstellungen und einer entsprechenden Hardwareausstattung empfehlen, das neue Potential in der einen oder anderen Form zu nutzen.

Literatur

Diekmann, Andreas; Manhart, Klaus, 1989:
Kooperative Strategien im Gefangenendilemma. Computersimulation eines N-Personen-Spiels,
In: Analyse & Kritik, 2, S.134-153.

Glas, Johann et al, 1992:
Windows 3.1 versus OS/2 2.0,
In: PC Professionell, 8, S.106-156.

Hüskes, Ralf; Shahrbabaki, Khatoun, 1993:
GUIs in Theorie und Praxis,
In: c't, 9, S.68-72.

Kühn von Burgsdorff, Bea, 1992:
Datenaustausch bequemer,
In: PC Magazin, 28, S.46-49.

Nitze, Michael, 1993:
Text und Formel,
In: c't, 1,S.108-114.

Microsoft, 1992:
Microsoft Windows Benutzerhandbuch, Version 3.1
Microsoft Corporation.

Winograd, Terry; Flores, Fernando, 1992:
Erkenntnis - Maschinen - Verstehen. Zur Neugestaltung von Computersystemen, Berlin.